

## EVALUATION OF QOS PARAMETERS FOR IPTV

Ancuta Sanda BUZILA, Gabriel LAZAR, Tudor BLAGA, Virgil DOBROTA  
*Technical University of Cluj Napoca, Communications Department*  
 George Baritiu 26-28, 400027 Cluj-Napoca, Romania, Tel: 40-264-401264, Fax: 40-264-597083  
 E-mails: anca\_buzila@yahoo.com, {gabriel.lazar,tudor.blaga,virgil.dobrota}@com.utcluj.ro

**Abstract:** This paper describes in details the IPTV system, together with its implementation methods and the involved protocols. Furthermore, measurement methods of QoS parameters are analyzed, discussing a practical application for evaluation of IPTV traffic parameters. The following components are taken into account: bandwidth, one-way delay, inter-packet jitter, packet loss, number of duplicated packets, out-of-order packets and corrupted packets. In addition, two QoS parameters of importance for VoD are measured: START delay and PAUSE/RESUME delays. The implementation of this program opens the possibility of measuring the transmission parameters that influence QoS, with the purpose of offering a way to evaluate and control the services delivered by open-standards IPTV providers.

**Key words:** IPTV, measurements, QoS, XML.

### I. INTRODUCTION

Nowadays there is a high trend towards the development and deployment of bandwidth consumer applications. In order to reduce cost new technologies, like IP telephony, IP videoconferencing and lately IP television (IPTV), are used. Telecom operators with large numbers of subscribers can not implement these types of applications in legacy, congested networks. The step to NGN (Next Generation Network) provides a robust and flexible environment better suited for the high performance requirements for future networks.

Known as IPTV, or IP television, this concept represents the convergence between broadcast, telecommunications and information technology. IPTV refers to digital television service and other audio and video services transmitted through broadband networks using the same protocol that sustains the Internet.

The IPTV concept includes both real time television and on demand television so called VoD (Video on Demand). Other services that IPTV can offer are Beyond TV (BTV) saving shows for later views, Time Shifted TV, on demand games and music, EPG (Electronic Program Guide), PVR functions with features similar to a DVD player, pay-per-view functions and parental control.

IPTV delivery implicates the existence of a media server for receiving the TV signal and transforming this signal into a digital signal that can be sent through a data network using the IP protocol at the third layer. A set top box can be used for receiving the data that will be transmitted to the TV set. A computer can also be used for receiving and displaying the media data.

IP multicast is used for delivering real time television to multiple clients at the same time. This type of delivery

reduces significantly bandwidth consumption [1].

As we have already mentioned IPTV refers to realtime televisions and also VoD, involving several protocols corresponding to different OSI layers, as in Figure 1. The video data is sent in MPEG-2 TS (Transport Stream) format using UDP with IP multicast. In this case IGMP (Internet Group Management Protocol) version two is used for multicast group management. In the case of VoD RTSP (Real Time Streaming Protocol) performs flow control for the transmission, allowing the user to start, pause/resume the stream. SSL (Secure Sockets Layer) provides security for IPTV [3].

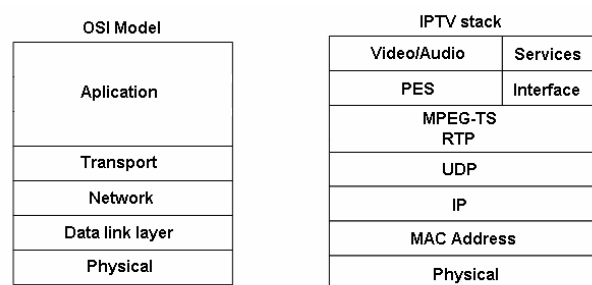


Figure 1. IPTV protocol stack for user data

In real time streaming data is sent using only the RTP (Real-time Transmission Protocol) protocol over UDP (User Datagram Protocol) protocol, as in Figure 1. The user can register or time shift the streamed data. Data is sent using MPEG TS protocol [4]. In addition to the RTP protocol the RTSP protocol is used for VoD streaming. This protocol is implemented over the TCP protocol and gives control over the streamed media to the client.

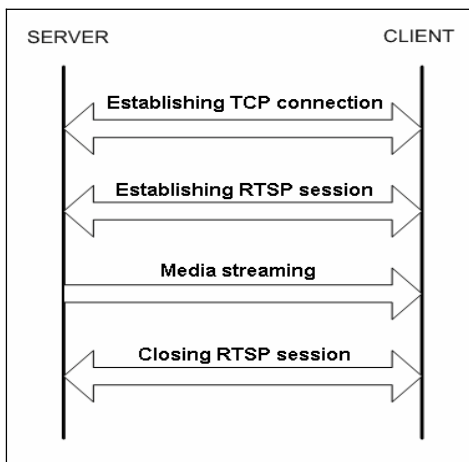


Figure 2. TCP/RTSP session

## II. QOS PARAMETERS FOR IPTV

For assuring a success for IPTV service operators must offer better quality than classic television. The distortion of the received IPTV signal is mainly caused by the variation of the one-way-delay parameter. A network capable of ensuring a level of QoS is the main step for offering the IPTV service.

QoS handles a set of parameters like bandwidth, one-way-delay, delay variation and packet loss when referring to a communication channel. Quality of service means the capacity of a network to offer better services for a selected part of the traffic. There are control mechanisms that offer different priorities for some users or for a part of the data traffic, or guarantees some quality levels depending on the application type [5]. The main interest of QoS is to assure a priority for band allocation, jitter and delay control, reducing the percentage of packet loss. For real time application like IPTV and VoIP a guaranteed QoS level is very important because this are CBR (Constant Bit Rate) applications and are delay sensitive. QoS technology brings new elements which will be used in all type of networks.

QoS parameters are the transmission parameters which affect the quality of a service, of the application that uses this service, even if it's only a data transfer or a real time streaming. QoS parameters for real time streaming are:

**Traffic rate:** Knowing the transmission rate we can verify if a network with a certain capacity can afford that type of transmission. For the calculus of the traffic rate we counted all the bytes sent and the duration of the transmission. Dividing the number of bytes by the transmission duration we obtain the traffic rate [6].

$$\text{Traffic\_rate} = \frac{\text{total\_bytes}}{\text{transmission\_duration}} [\text{Bps}] \quad (1)$$

**One-way-delay** is the delay introduced by the transmission of packets from source to destination. This parameter depends on many elements like number of nodes to pass toward destination, network traffic, routing protocols. For measuring the one-way-delay parameter it is important to assure the synchronization between the server and the receiver. If the two machines are not synchronized the parameter computed is not correct.

$$\text{one-way-delay} = \frac{\text{delay}_1 + \text{delay}_2 + \dots + \text{delay}_N}{N} [s] \quad (2)$$

**Jitter:** The packets sent can arrive to destination following different paths so the delay of the packets can differ. Jitter is the variation delay of packets and it is a very important parameter for real time streaming [7]. Inter packet jitter is defined within the Formula (3).

$$\text{Jitter} = \frac{|\text{jitter}_1| + |\text{jitter}_2| + \dots + |\text{jitter}_N|}{N} [s] \quad (3)$$

**Number of lost packets:** In its path from source to destination a packet can be lost or eliminated by a router if the router's buffer is full or if the packet is corrupted. The elimination of packets depends only on the current state of the network, and it cannot be anticipated. The algorithm used for determining this parameter identifies each packet sent by the server and searches the packet in the received packet list.

**Number of reordered packets:** When more packets are sent in a certain order by an application, at the receiving side they may arrive out-of-order due to possible different paths chosen by routers. At the destination other protocol must be used for reordering the packets. This process is very important especially for video transmissions and VoIP applications, where the quality is most affected by the delay. For counting the number of reordered packets it is necessary to identify every packet sent and received. A definition for packet reordering is found in [8]: a packet is considered reordered if the sequence number is smaller than the sequence number of the previous packet received. We use the RTP sequence number of the packets for every UDP port used during transmission.

**Number of duplicated packets:** Counting the number of packets duplicated is a way of verifying the network configuration. When duplicated packets appear this means that there are some configuration errors in the network or some devices are defective. The algorithm implemented is similar to that one used for counting the reordered packets.

**Number of corrupted packets:** When transmitting packets through the network some bits can be corrupted. These errors do not affect too much the quality as long as the number of corrupted bits is low. We can calculate the PER (Packet Error Rate) as in Formula (4). To see if a packet is corrupted or not we compare the data field of every packet at the sender and the receiver.

$$\text{PER} = \frac{\text{number\_corrupted\_packets}}{\text{number\_received\_packets}} \times 100\% \quad (4)$$

**START delay for VoD streaming** is measured at the receiver (i.e. the client) and represents the duration between the moment the first TCP packet, containing signaling, is sent by the client until the moment the first RTP packet, issued by the server and containing requested data, is received at the destination. This parameter is computed only for VoD transmissions.

**PAUSE/RESUME delay for the VoD server:** VoD transmission uses RTSP that gives stream control to the receiver. This can suspend the stream and start it back later or it can move forward and backward within media stream, as in Figure 3.

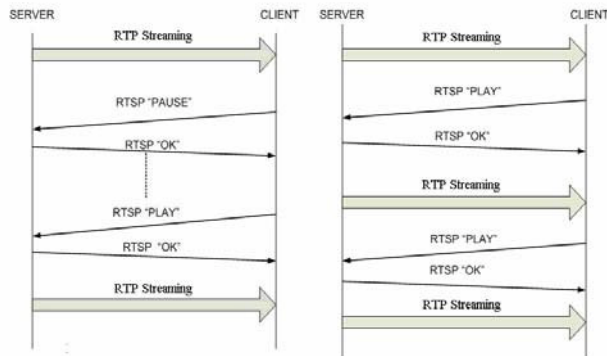


Figure 3. Examples of RTSP sessions

### III. IPTV TESTBED ARCHITECTURE

At least two elements are necessary for an IPTV transmission, a media server and a receiver. Both the client and the server were implemented using VLC [10]. The VideoLAN Server sends data using protocols and codecs for IPTV streaming. A router is needed to connect the server and the client. Queuing disciplines can be implemented for simulation of different network problems, like congestion or packet loss.

Figure 4 presents a minimal network architecture that involves a media server, an IPTV client and a router.

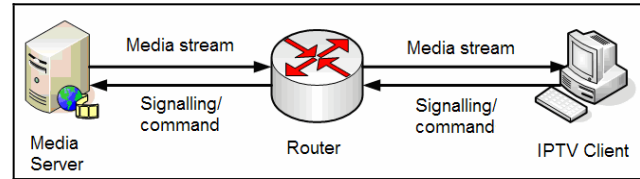


Figure 4. Testbed

An application in C++ was implemented in order to measure the QoS parameters. The program `iptv_tool` was compiled under Linux using `gcc`. Two arguments are necessary for running the program, the XML files containing the packets captured by Wireshark on the server's interface and on the client's interface. For each selected packet all fields are displayed up to the last recognized protocol. The capture file can be exported in XML format and can be parsed more easily than the original .pcap format.

The C++ functions defined within the library `libxml/xmlreader.h` were used for reading the value of the attributes from the XML file. This information, for example the total size of the packet or the timestamp, is saved and is then used for packet identification and calculus of the QoS parameters. Note that only the packets belonging to the IPTV stream are analyzed and included into a list. These packets are identified by the IP source and destination address and by the protocols and ports used for the transmission. The value of the attribute timestamp is read for every packet and saved into the current packet included into the list. The size of the packet is obtained from the attribute "frame size" field. The measurement of the one way delay parameter implies very good clock synchronization between the server and the receiver, because the delay is obtained by comparing the timestamps of the sent and received packet. Any synchronization problem between the two leads to erroneous values. This issue was eliminated by using a tweaked testbed Figure 5, compared to the one presented in Figure 4.

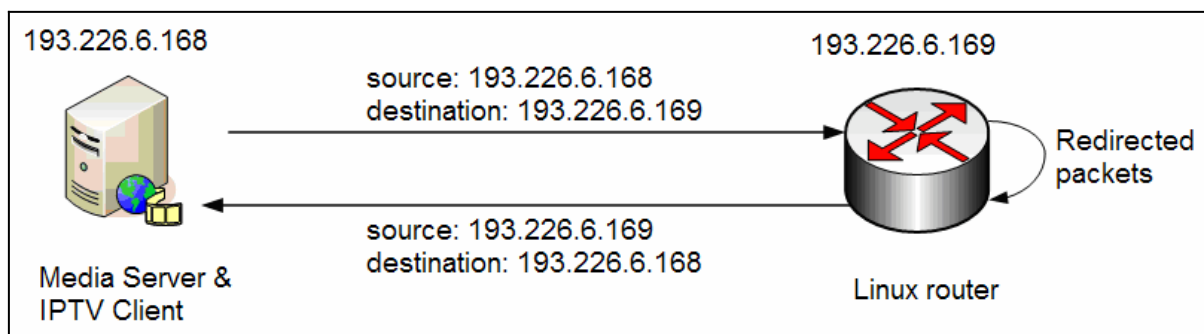


Figure 5. Testbed

We have the same entities, but in this case the server and the client are running on the same computer. The router has to redirect the packets received from the media server to the IPTV client that is located on the same interface. This is performed by switching the source IP address with the destination IP address. The tool used is `iptables` that allows packet filtering, manipulation and performs NAT (Network Address Translation). A special rule was inserted in the NAT table. For all the packets from PREROUTING table with the source address 193.226.6.168, the destination address is changed to the source address. In the

POSTROUTING chain the source address is modified to 193.226.6.169, the IP address of the router. The following rules were used:

```
#iptables -t nat -A PREROUTING -s \
193.226.6.168 -j DNAT--to- destination \
193.226.6.169
#iptables -t nat -A POSTROUTING -s \
193.226.6.168 -j SNAT --to- source \
193.226.6.169
```

When packet capture is performed at the server/client we

must be able to differentiate the sent and received media stream. Every stream is identified by the source and destination addresses. A feature of Wireshark performs filtering according to the `ip.src` and `ip.dst` parameters, thus the traffic can be saved into different files.

In order to introduce delay and packet loss, we used the `tc` program, which is part of the `iproute2` toolkit. It allows the modification of kernel QoS parameters. We can control the queuing discipline involved by `qdisc` and with the aid of `netem` we can specify the delay, packet loss, packet duplication and reordering.

One broadcast element was created in the user friendly VLM interface for testing the IPTV transmission. This is available only from the HTTP interface and supports remote connections. MPEG2-encoded video sequence, at the rate of 2048 kbps, has been used for testing real time IPTV streaming. The audio codec used is MP3 and then all is packed in MPEG1 format.

For testing the VoD transmission we created a VoD element in the VLM interface and a H.264 encoded video sequence at the rate of 3072 kbps has been used. The audio codec used is MPEG4, all being packed in MP4 format.

Several test scenarios were employed. In the first one the packets were redirected, without any further delay or loss. `iptv_tool` was used to display the determined parameters in `xgraph` format. Further scenarios with delay, loss, packet duplication, packer corruption and reordering were also considered (as in Table 1), either for realtime streaming or for VoD.

Table 1. Realtime streaming/VoD test scenarios

	1	2	3	4	5	6	7
delay [ms]	0	100	0	0	0	0	100
jitter [ms]	0	10	0	0	0	0	10
loss	0	0	5%	0	0	0	5%
error	0	0	0	20%	0	0	0
duplication	0	0	0	0	0	10 %	5%
reordering	0	0	0	0	25%	0	10%

#### IV. EXPERIMENTAL RESULTS

##### IV.a. REALTIME STREAMING

The first test for the IPTV transmission we used the script `router` for simple redirecting of the packets. The results confirmed that for real time streaming only RTP packets are sent without any previous connection. All the packets sent were received at the user. In Figure 6 we can see how the throughput at the server and the receiver are equal during the entire transmission, the average transmission rate obtained being 327.075 kbps. The one-way-delay parameter can be observed in Figure 7, the maximum value obtained is 1.9 ms and the minimum value is 0.555 ms. The medium value of the one-way-delay is of 0.569 ms.

A very important QoS parameter is jitter and you can observe this parameter in Figure 8. We can observe that the jitter has very low values; the medium value obtained is 0.014 ms with only five spins of higher values during the transmission.



Figure 6. Test1-IPTV Traffic rate sent from the server and received at the client

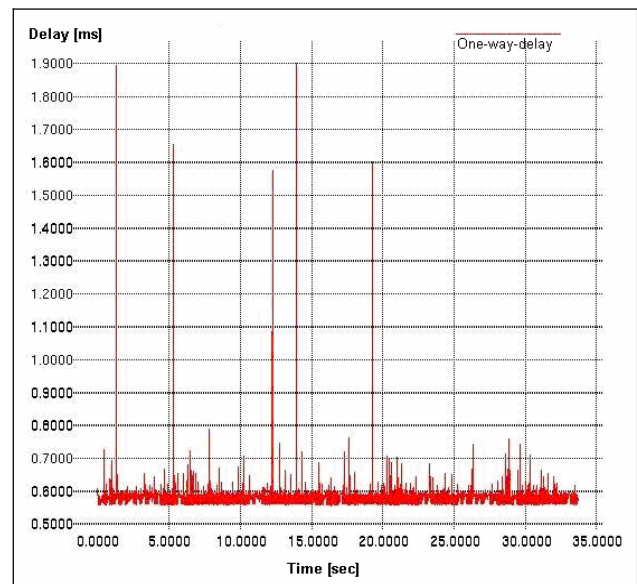


Figure 7. Test1-IPTV One-way delay

The maximum value of the jitter parameter is 1.3 ms whilst the medium value was computed using the absolute value of the inter-packet jitter. The values obtained for the other parameters are zero because no distortion was introduced during this transmission.



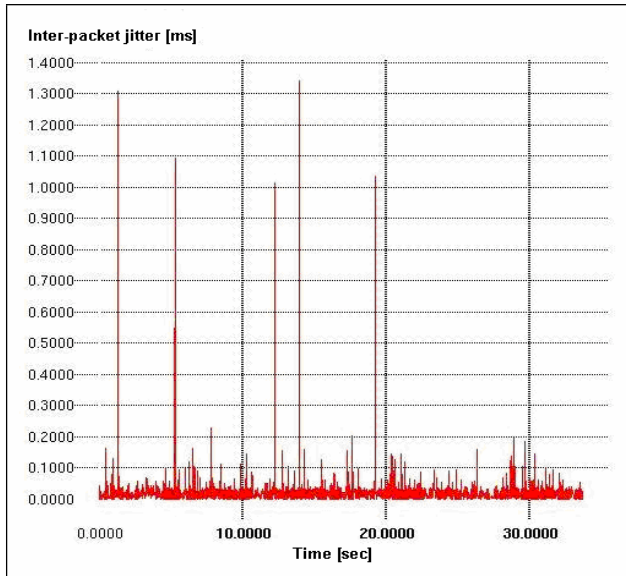


Figure 8. Test1 - IPTV Inter-packet jitter

For the second test the script `router_delay` was used: this script introduces a delay of 100 ms and a jitter between -10 ms and +10 ms for every packet that passes through the router.

After running the application `iptv_tool` we see that the only differences to the previous test are in one-way-delay parameter, jitter and packet reordering. The transmission rates are the same for the server and for the receiver. We can see in Figure 9 that the one-way-delay parameter is with 100 ms bigger than at the previous test. The value obtained now was 102.698 ms. This difference is caused by the delay introduced by the router

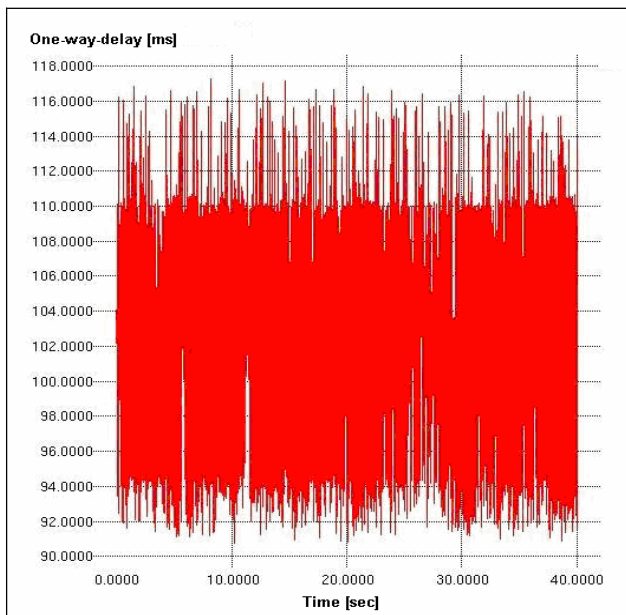


Figure 9. Test 2-IPTV One-way-delay for RTP packets

The jitter resulted after this second test is bigger than that one obtained within the first test. The difference is about 6.6 ms and represents the medium value of the inter packet jitter

introduced by the router. The big value of the jitter parameter is causing a reordering of the packets at the receiver so now we get a percentage of 53% packet reordering.

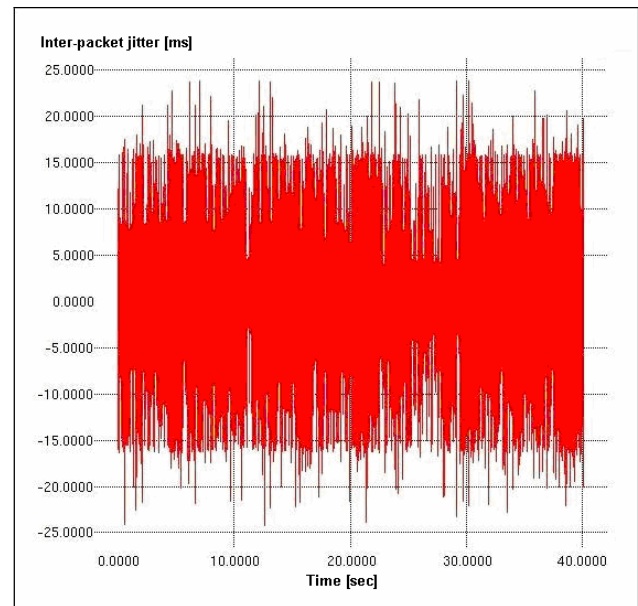


Figure 10. Test 2-IPTV Inter-packet jitter

#### IV.b. VIDEO on DEMAND

For the first test of the IPTV transmission we use the `router` script.

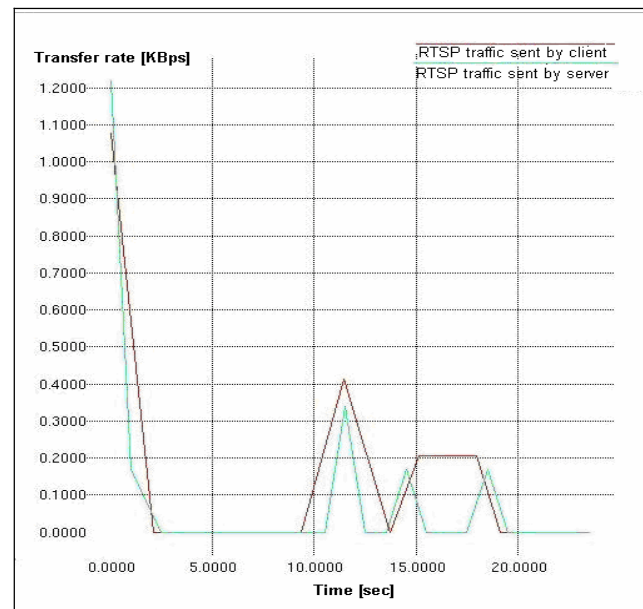


Figure 11. Test 1-VoD Transfer rate for RTSP session

For VoD transmission more parameter are calculated and displayed because this type of transmission uses more protocols. Besides the parameters measured for the IPTV transmission there are two more calculated: the START delay and the PAUSE/RESUME delay for VoD streaming.

The value of the one-way-delay parameter is 0.571 ms a close value to the one obtained for the IPTV transmission

using the same script. The jitter is 0.072 ms, also a close value to the previous obtained at IPTV testing.

An interesting graphic to see is that one displaying the RTSP traffic sent both by the server and the client, as in Figure 11. The medium value of the PAUSE/RESUME delay parameter obtained is of 342.045 ms and represents the media of the intervals between the moment the RTSP Play packets were sent and the moment the RTSP Ok packets arrived. We can observe how fast the server responds to the client requests during the transmission. The START delay parameter is of 2.4 s and it has a major contribution to the delay of the VoD transmission. This is due to the TCP protocol used to transmit the RTSP messages.

For the second test we used the `router_delay` script as for the corresponding one in the previous paragraph. The resulted displayed, i.e. 102.376 ms, shows a higher value for the one-way-delay parameter, greater than the value obtained during first test with about 100 ms. Note that this is exactly the delay introduced by the router.

In Figure 12 we can see the PAUSE/RESUME delay during the transmission. The medium value of the parameter is 449.504 ms, being with 100 ms greater than the value for the previous test. The START delay parameter obtained was 3.519s, with about 0.9 s greater. We can see that the delay introduced by the router affects the one-way-delay, the START delay and the PAUSE/RESUME delay. This also causes about 60 % of reordered packets. Other five scenarios were analyzed and the resulting parameters confirmed the good performances of the `iptv_tool`.

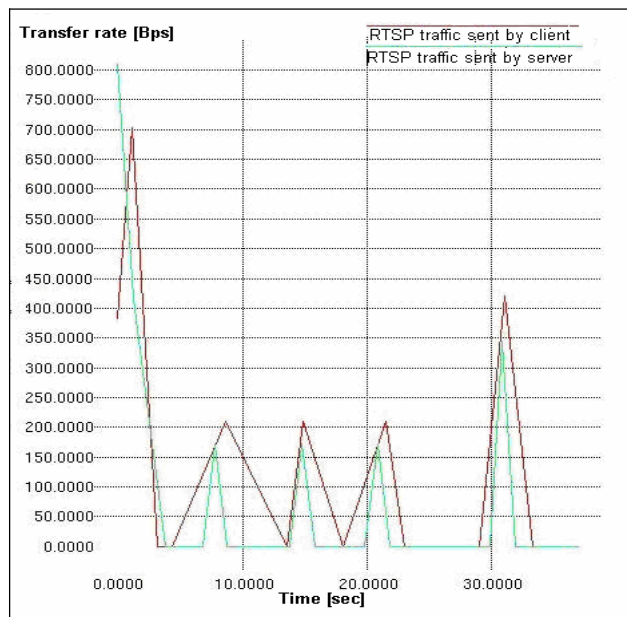


Figure 12. Test 2-VoD Transfer rate for RTSP session

## V. CONCLUSIONS AND FURTHER WORK

We described the functionality of an IPTV system, the protocols involved in this type of transmission and the implementation methods. Furthermore measurements of QoS parameters were analyzed, and the results made possible a practical application to evaluate the IPTV traffic parameters. The following components are taken into account: bandwidth, one-way-delay, inter-packet jitter, packet loss, number of duplicated packets, out-of-order packets and corrupted packets. In addition, two QoS

parameters of importance for VoD were measured: START delay and PAUSE/RESUME delay. We used a tweaked testbed for eliminating the synchronization problem between the sender and the receiver. In this paper we presented a software tool that can measure the QoS parameters for real time streaming IPTV and VoD. The implementation of this program opens the possibility of measuring the transmission parameters that influence QoS, with the purpose of offering a way to evaluate and control the services delivered by open-standards IPTV providers. All tests confirm the functionality of this application.

Further work can be done for establishing the QoE (Quality of Entertainment) parameters using the tool implemented.

## REFERENCES

- [1] W.Coopor, G.Lovelace, "IPTV Guide: Delivering Audio and Video over Broadband", *IPTV Report*, 2006. <http://iptv-report.com/guide/request/download/IPTV-Guide.pdf>
- [2] V.Dobrota, *Digital Networks in Telecommunications: Volume III OSI and TCP/IP*, Second Edition, Mediamira Science Publishers, Cluj-Napoca 2003.
- [3] \*\*\*, "Aptixia IxLoad: Video- IGMP, MPEG, RTSP and RTP", *Ixia* 2007  
[http://www.ixiacom.com/products/display?skey=aptixia\\_ixload\\_igmp\\_mpeg\\_rtsp\\_rtp](http://www.ixiacom.com/products/display?skey=aptixia_ixload_igmp_mpeg_rtsp_rtp)
- [4] Y. Kikuchi et al., "RTP Payload Format for MPEG4 Audio/Visual Streams", *RFC 3016*, November 2000
- [5] G.Armitage, *Quality of Service in IP Networks*, New Riders Publishing, 2000
- [6] N.Brownlee, "Traffic Flow Measurement: Architecture", *RFC 2063*, 1997.
- [7] C.Demichelis, P.Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics", *RFC3393*, 2002
- [8] V.Paxson, "Framework for IP Performance Metrics", *RFC2330*, 1998
- [9] \*\*\*, "IPTV Test Scenario: "Easy-to-Configure IPTV/ Triple Play Testing", *Spirent Communications*, 2006
- [10] \*\*\*, "VLC - The Cross-Platform Media Player and Streaming Server", [www.videolan.org/vlc/](http://www.videolan.org/vlc/)
- [11] \*\*\*, Wireshark, <http://www.wireshark.org/>